EasyChair Preprint

№ 13243

# Temporal Knowledge Graph Link Prediction Using Synergized Large Language Models and Temporal Knowledge Graphs

Yao Chen and Yuming Shen



May 18, 2024

# Temporal Knowledge Graph Link Prediction using Synergized Large Language Models and Temporal Knowledge Graphs

Yao Chen[1] Yuming Shen[1*]

[1] School of Information Science and Technology,
Guangdong University of Foreign Studies, China

ymshen2002@163.com

**Abstract.** Although large language models and temporal knowledge graphs each have significant advantages in the field of artificial intelligence, they also face certain challenges. However, through collaboration, large language models and temporal knowledge graphs can complement each other, addressing their respective shortcomings. This collaborative approach aims to harness the potential feasibility and practical effectiveness of large language models as external knowledge bases for temporal knowledge graph reasoning tasks.In our research, we have meticulously designed a synergized model that leverages the knowledge from the graph as prompts. The answers generated by the large language model undergo careful processing before being seamlessly incorporated into the training dataset. The ultimate goal is to significantly enhance the reasoning capabilities of temporal knowledge graphs. Experimental results underscore the positive impact of this synergized model on the completion tasks of temporal knowledge graphs, showcasing its potential to address gaps in knowledge and improve overall performance. While its influence on prediction tasks is relatively weak, the collaborative synergy demonstrates promising avenues for further exploration and development in the realm of AI research.

**Keywords:** Large language models, Temporal knowledge graphs, Synergetic pattern, Completion task, Prediction task.

## 1 Introduction

The concept of knowledge graphs can be traced back to the Semantic Net in 1956, and it gained renewed attention in 2012 when Google officially introduced the term "knowledge graph" . Various knowledge graphs, such as Google Knowledge Graph, financial knowledge graphs, and legal knowledge graphs, have been developed, enhancing search engine effectiveness and contributing to the development of intelligent assistants. Knowledge graphs are essentially graph databases storing information in triplets [entity, relation, entity] or [entity, attribute, attribute value]. For

example, [Yaoming, was born in, Shanghai] represents a relationship between the entity "Yaoming" and "Shanghai."

Traditional knowledge graphs primarily describe static common knowledge, often neglecting temporal information. In 2018, Leblay et al. introduced temporal knowledge graphs [1], incorporating temporal concepts into traditional knowledge graphs with a four-tuple format, including a time attribute. Temporal knowledge graphs consider temporal information, effectively managing dynamically evolving temporal knowledge and supporting time-coupled applications. For instance, [LeBron James, played for, Cleveland Cavaliers, 2003 to 2010] and [LeBron James, played for, Los Angeles Lakers, 2018 to present] reflect changes over time.

However, real-world temporal knowledge graphs cannot cover all knowledge, leading to significant knowledge gaps affecting downstream application performance. For example, knowledge base question-answering systems struggle with queries about information not present in the knowledge base, requiring inference models to automatically discover missing knowledge. Additionally, downstream applications need to predict future knowledge, such as e-commerce recommendation systems suggesting the next possible purchase for users and medical decision support systems predicting disease development based on patient historical clinical data. Therefore, knowledge inference tasks are crucial for temporal knowledge graphs, involving completion and prediction tasks. Completion utilizes historical and future knowledge relative to the knowledge to be completed, emphasizing bidirectional time characteristics, while prediction relies on historical knowledge, using time unidirectional feature learning for evolution patterns.

Large Language Models (LLMs) represent a significant breakthrough in artificial intelligence, with parameters reaching billions of weights. Trained on millions to billions of text data, LLMs can adapt to various contexts, generating more natural language in tasks such as text summarization, question-answering systems, and machine translation. However, LLMs, as parameterized implicit knowledge, face challenges like factual fabrication and lack of interpretability. Temporal Knowledge Graphs (TKGs), as structured explicit knowledge, offer natural interpretability and demonstrate high-quality knowledge representations in specific domains. Although TKGs have higher construction costs, may not be exhaustive, and lack natural language processing capabilities, their structured features aid in a deeper understanding of model outputs. To overcome individual deficiencies and leverage complementary advantages, LLMs and TKGs can collaborate, enhancing natural language processing capabilities and expanding application scope. Specifically, TKGs can provide additional, structured, high-quality knowledge to LLMs, improving model generalization. Simultaneously, LLMs can automatically extract knowledge from text data, reducing TKG construction and maintenance costs, making it more comprehensive. The synergy between TKGs and LLMs is still in its early stages and holds significant research value.

The large language models can be classified into open-source and closed-source categories. This paper utilizes the closed-source large language models, and its main contributions are as follows:(1)We introduce a novel collaborative mode between large language models and temporal knowledge graphs.(2)This collaborative

approach has a positive impact on the task of completing temporal knowledge graphs, leading to significant improvements in TeLM's inference performance. The MRR metric shows performance improvements of 3.2% and 2.7% on the Yago11k and Wiki12k datasets, respectively.

## 2 Related Work

### 2.1 Static Knowledge Graph

Various methods have been developed to model static Knowledge Graphs (KGs) without incorporating temporally dynamic facts, as summarized in recent surveys [2]. One category of these methods includes translational models [3], which represent the relation between two entities as a translation vector. Another category involves semantic matching models that assess the plausibility of facts using a triangular norm [4]. Additionally, there are models based on deep neural network approaches, utilizing feed-forward or convolutional layers on top of embeddings [5]. However, these approaches do not account for temporally dynamic facts.

### 2.2 Temporal Knowledge Graph

Recent efforts have aimed to capture the evolving nature of facts in Temporal Knowledge Graphs (TKGs). TTransE [6] extends TransE [7] by incorporating temporal information into the score function. HyTE [8] introduces a time-specific normal vector in place of the unit normal vector in the hyperplane projection of TransH [9]. Know-Evolve [10] focuses on learning non-linearly evolving entity representations over time, treating the occurrence of a fact as a temporal point process.

Models like ConT [11], based on the Tucker decomposition, learn a new core tensor for each timestamp but lack a mechanism to capture long-term dependencies in consecutive time snapshots.

Other methods are designed to model graph sequences for capturing the long-term dependency of TKG facts. TA-DistMult [12] uses a recurrent neural network to learn time-aware representations of relations, employing standard scoring functions from DistMult [13]. GCRN [14] combines Graph Convolutional Networks (GCN) for graph-structured data with RNN to identify meaningful spatial structures and dynamic patterns simultaneously. DyREP [15] divides the dynamic graph network into two processes: global and local topological evolution. It proposes a two-time scale deep temporal point process to model these processes jointly. Know-Evolve, DyREP, and GCRN have also been integrated with MLP decoders for predicting future facts, as demonstrated by Jin. A previous state-of-the-art method in this research line, RE-NET [16], jointly models event (fact) sequences using an RNN-based event encoder and an RGCN-based [17] snapshot graph encoder.

### 2.3 Large Language Models

The application scope of large language models is increasingly extensive, making it essential to comprehensively understand and effectively evaluate their various capabilities. This is crucial for designing new paradigms for human-machine interaction. For natural language processing tasks, these capabilities encompass sentiment analysis [18], text classification [19], natural language inference [20], question-answering systems [21], natural language generation [22], authenticity verification [23], and the ability to handle multilingual tasks [24].

### 2.4 Synergized LLMs and KGs

In recent years, the collaborative interaction between Large Language Models (LLMs) and Knowledge Graphs (KGs) has gained increasing attention. The fusion of the strengths of LLMs and KGs aims to mutually enhance performance in various downstream applications, primarily manifested in the following three aspects.

Firstly, Collaborative Knowledge Representation of LLMs and KGs involves combining Large Language Models (LLMs) with Knowledge Graphs (KGs) to jointly represent and utilize information from both textual corpora and structured knowledge. A representative method, BERT-MK [25], employs a dual-encoder architecture, enhancing the knowledge encoder component with additional information about adjacent entities during the pre-training phase of LLM. However, certain adjacent entities in the knowledge graph may not have direct relevance to the input text, introducing unnecessary redundancy and noise. Therefore, CokeBERT [26] proposes an innovative approach, introducing a Graph Neural Network (GNN)-based module capable of intelligently filtering out irrelevant KG entities based on the input text. Simultaneously, JAKET [27] suggests another strategy, integrating entity information in the intermediate stage of the large language model to optimize the knowledge integration process.

Secondly, Collaborative Reasoning of LLMs and KGs involves inferring knowledge by combining the sources of knowledge from Large Language Models (LLMs) and Knowledge Graphs (KGs). To enhance interaction between text and knowledge, KagNet [28] proposes a strategy of first encoding the input KG and then using the encoded KG information to augment the representation of the input text. In contrast, MHGRN [29] adopts a different approach, using the final output of the LLM to guide the reasoning process on the KG. However, both these methods only design unidirectional interaction between text and KG, which limits the effectiveness of knowledge fusion.

To address this limitation, QA-GNN [30] introduces a Message Passing Graph Neural Network (GNN) model for joint reasoning on input context and KG information obtained through message passing. Specifically, QA-GNN represents input text information as a special node and connects this node with other entities in the KG through pooling operations. However, a limitation of this approach is that the text input is aggregated into a single dense vector, potentially restricting the performance of information fusion.

To further enhance the interaction between text and KG, JointLK [31] proposes a novel framework. This framework achieves fine-grained interaction between any token in the text input and KG entities through a bidirectional attention mechanism between LM-to-KG and KG-to-LM. In contrast, GreaseLM [32] designs deep and rich interactions between input text tokens and KG entities at each layer of the LLM. This design allows GreaseLM to better leverage complementary information between text and KG during the reasoning process, thereby improving the accuracy and efficiency of reasoning.

## 3 Methodology

In this section, we will introduce the detailed overview of the collaborative framework between the Large Language Models (LLMs) and the Temporal Knowledge Graphs (TKGs) designed in this paper.Figure 1 shows the synergized LLMs and TKGs.

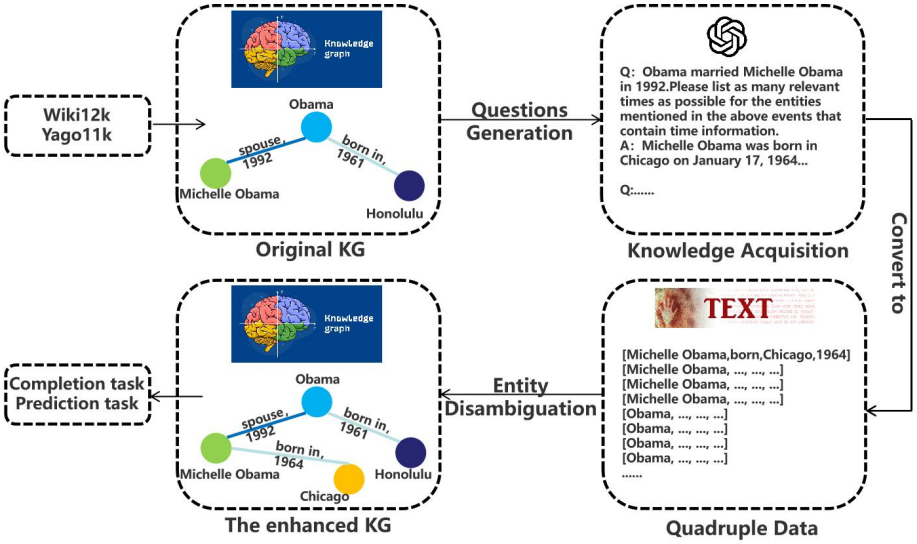


**Fig. 1.** Synergized LLMs and TKGs.We convert the structured data from the temporal knowledge graph datasets Wiki12k and Yago11k into natural language sentences to construct a large language model question-answer dataset. Then, we process the responses returned by the large language model and add the data to the training set of the inference task.

### 3.1 Construct the LLM Question-and-answer Dataset

In this paper, we primarily utilized two temporal knowledge graph reasoning task datasets: Wikidata12k [33] and YAGO11k [34].

Wikidata12k is a temporal knowledge graph dataset constructed based on Wikipedia, selecting the top 24 most frequent relations from the open knowledge

graph Wikidata. The dataset spans from the year 1709 to 2018, comprising 12,554 entities, 24 relations, and 40,621 relation facts. Encompassing various domains such as history, science, and culture, this dataset provides detailed temporal information, offering researchers a rich resource for in-depth exploration and experimentation in temporal knowledge graph reasoning tasks.

YAGO11k, derived from the YAGO (Yet Another Great Ontology) project, is a dataset selected from the temporal knowledge of the open knowledge graph YAGO3, focusing on the top 10 most frequent relations. The temporal scope of this dataset is more extensive, ranging from 453 BC to 2844 AD. With 10,623 entities, 10 relations, and 20,509 relation facts, the dataset covers various entity types, including persons, organizations, and locations, accompanied by temporal annotations. It serves as a valuable resource for researchers to explore temporal relationships among entities.Table 1 summarizes the statistics of the datasets.

**Table 1.** Statistics of the datasets.

| #Data | #Entities | #Relations | #Time Span | #Training | #Validation | #Test |
|---|---|---|---|---|---|---|
| Wiki12k | 12,554 | 24 | 1709-2018 | 32,197 | 4,062 | 4,062 |
| Yago11k | 10,623 | 10 | -453-2844 | 16,408 | 2,050 | 2,051 |

We need to transform the structured data of these quadruples into natural language sentences. As the textual information in the data typically consists of simple keywords for entities and relationships, it is insufficient to fully describe an event. Therefore, based on the characteristics of the data, we need to convert relationship nouns into appropriate predicate verbs and add prepositions as needed to form complete natural language sentences. For example, the quadruple [Obama, place of birth, Hawaii, 1961] is transformed into the natural language sentence "Obama was born in Hawaii in 1961."

In addition, we also need to handle time attributes, which can be classified into four types: time points, time intervals with both start and end times, time intervals with only end times, and time intervals with only start times. For the first type, i.e., time points, such as [Obama, place of birth, Hawaii, 1961], the transformation into a natural language sentence would be "Obama was born in Hawaii in 1961." For the second type, i.e., time intervals with both start and end times, like [LeBron James, plays for, Cleveland Cavaliers, 2014, 2018], the transformation would be "LeBron James played for the Cleveland Cavaliers from 2014 to 2018." The third type of time interval only includes an end time, as in [Hans Helfritz, country of citizenship, Germany, ####, 1948], and the transformation would be "Hans Helfritz's nationality was Germany until 1948." Lastly, the fourth type of time interval only includes a start time, such as [Oleh Mishchenko, member of sports team, FC Amkar Perm, 2016, ####], and the transformation would be "Oleh Mishchenko has been a member of the FC Amkar Perm since 2016."

The example is as follows:Obama(head entity) was born(relation) in Hawaii(tail entity) in 1961(time).Please list as many relevant events as possible that contain time information for the entities or relation mentioned in the above events.

### 3.2 Knowledge Acquisition

We utilize large language models such as ChatGPT 3.5, Bard, Claude, and ERNIE Bot for knowledge acquisition.The next step is the design phase for prompts. Our goal is to guide the large language model to provide as many event details related to the entities mentioned in the previous events and rich in time information.The answers are typically in natural language by default. We can save these answers and then, with the help of a large language model, extract structured quadruple data from the unstructured natural language data. However, this undoubtedly adds a lot of workload. Therefore, we hope the answers are in the form of structured quadruple data. Figure 2 shows the Large Language Model Knowledge Acquisition.

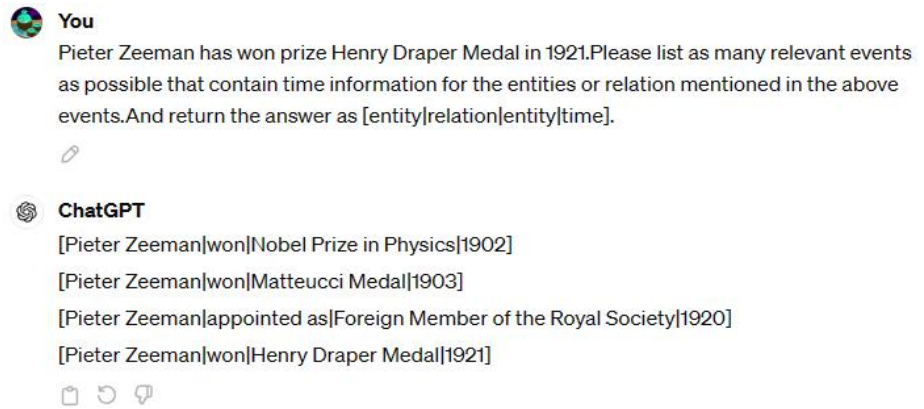


**Fig. 2.** The Large Language Model Knowledge Acquisition.We use the relevant information of Pieter Zeeman in the question-and-answer dataset as a prompt and design prompts to retrieve more related quadruple information through large language models.

### 3.3 Data cleaning

Due to the diverse formats of answers generated by large language models, they do not always conform to standard structured data formats. To ensure the integrity, accuracy, consistency, and usability of the data, we need to perform meticulous identification, modification, or removal operations on these data. The entire data cleaning process includes the following key steps:

**Extracting Quadruple Data.** Quadruple data generated by large language models often use symbols such as "[]", "**", "()", etc., for enclosing, while entities, relations, times, and other elements are typically separated by "|" or ",". In standard quadruple data, you would usually see three "|" or three ",". For those with fewer than three "|" or ",", their structure is incomplete, and we can directly delete them.

**Removing Invalid Data.** Due to the incomplete nature of answers generated by large language models, we need to remove quadruples with missing data, specifically those with missing entities or time attributes. Examples of missing data include "| - |", "unknown", "N/A", "?", "None", etc. To handle these incomplete data, we can read quadruple data line by line and use string matching to identify and delete entire lines containing any of the mentioned strings.

**Performing Data Type Conversion.** For explicit time ranges like "[1996 to 2024]", we can directly convert it to the format "[1996-##-##\t2024-##-##]". As for time representations like "[before 1996]" and "[after 1996]", we can respectively transform them into the formats "[####-##-##\t1996-##-##]" and "[1996-##-##\t####-##-##]".

**Eliminating Duplicate Values.** For duplicate answers, we need to remove them, and the set() data structure in Python can efficiently accomplish this task. By converting the data into a set, we can ensure that there are no duplicate elements in the dataset.

### 3.4 Entity disambiguation

Entity disambiguation is a crucial task in natural language processing, aiming to address the issue where the same entity has multiple surface forms (such as names, spelling variations, abbreviations, etc.) in different contexts. In texts generated by large language models, this ambiguity is particularly pronounced due to the lack of specific contextual knowledge and standardized datasets. To tackle this problem, knowledge graphs like Wikidata can be utilized for entity linking and disambiguation.

## 4 Experiments

We leverage advanced large language models such as ChatGPT 3.5, Bard, Claude, and ERNIE Bot to acquire rich new knowledge. Subsequently, we integrate this valuable knowledge into the original Wiki12k and Yago11k datasets to effectively enhance the data. Building upon this, we employ temporal knowledge graph completion tasks and prediction tasks to validate the practical application value of the newly added knowledge. Finally, by comparing the model inference performance using the augmented dataset with the original dataset, we can assess the extent to which this new knowledge improves the model's performance.

### 4.1 Datasets

The two datasets used in this paper are enhanced versions of Wiki12k and Yago11k. Wiki12k represents the original dataset, while GPT-Wiki, Bard-Wiki, Claude-Wiki, and ERNIE-Wiki represent Wiki12k datasets that have undergone data enhancement using large language models ChatGPT 3.5, Bard, Claude, and ERNIE, respectively.

The processing approach for the Yago dataset is similar.Figure 3 shows the statistics of the training data.

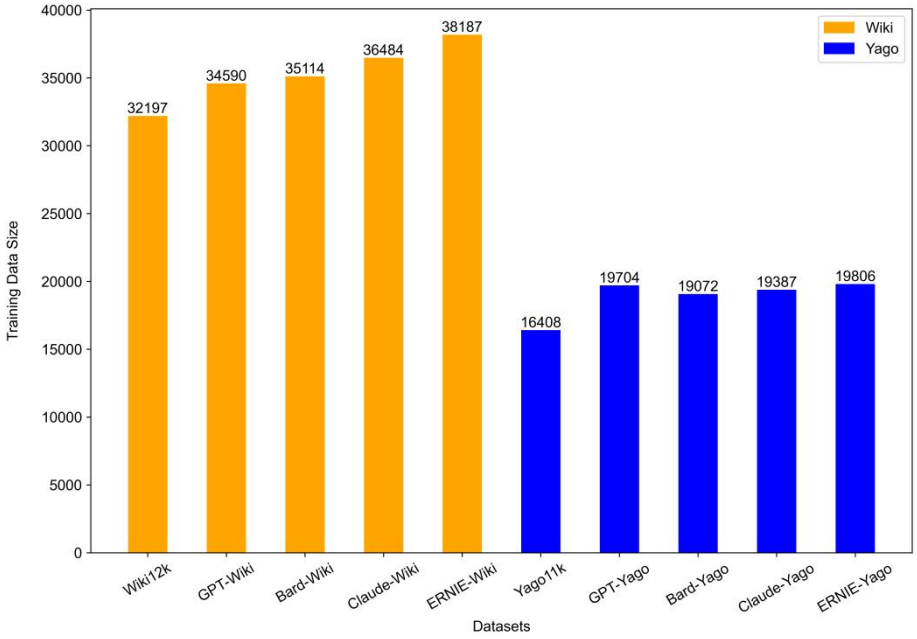


**Fig. 3.** The Statistics of Training Data.The data volume of ERNIE-Wiki and ERNIE-Yago is the highest, with an increase of 5990 and 3398 respectively compared to the original dataset.

### 4.2 Evaluation Protocols

In this context, two evaluation metrics are employed: Mean Reciprocal Rank (MRR) and Hits@k. Mean Reciprocal Rank is calculated as the mean of the reciprocal values of all computed ranks. Hits@k represents the fraction of test quadruples that rank within the top k.

### 4.3 Model Configurations

We validated the impact of this collaborative mode on the performance of the temporal knowledge graph completion task using the ATiSE [35] and TeLM[36] models. Additionally, we employed the CyGNet [37] and RE-Net [16] models to assess the effect of this collaborative mode on the performance of the temporal knowledge graph prediction task.The hyperparameters for the above-mentioned models are set based on the optimal values provided in the respective papers.

# 5 Results and Analysis

## 5.1 Completion Tasks

This task employed the ATiSE model and TeLM model. Tables 2 and 3 respectively describe the results of TeLM and ATiSE on the Yago11k and Wiki12k datasets. In these experiments, Claude, Bard, ERNIE, and GPT-3.5 represent the names of the datasets enhanced by the corresponding large language models, while "Origin" denotes the original dataset.

From the results, it is evident that the dataset enhanced by the large language model Bard achieved optimal performance for both the TeLM and ATiSE models.On the other hand, the dataset enhanced by the large language model ERNIE did not perform as well on the ATiSE model.Overall, the dataset enhanced by the large language models showed a significant improvement in performance on the TeLM model.

The improvement in the performance of the completion model demonstrates that the majority of the data generated by the large language model is accurate. When this data is added to the training set of temporal knowledge graph reasoning tasks, it provides the model with more effective language information and background knowledge. This aids the model in better understanding and memorizing information related to time.

**Table 2.** TeLM results on Yago11k and Wiki12k.

| Method | Yago11k | | | | Wiki12k | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| Claude | .195 | .134 | .194 | .326 | .340 | .239 | .372 | .556 |
| Bard | **.220** | **.161** | **.224** | **.344** | **.353** | **.251** | **.387** | **.570** |
| ERNIE | .180 | .118 | .180 | .314 | .333 | .231 | .367 | .545 |
| GPT-3.5 | .195 | .131 | .198 | .331 | .330 | .231 | .360 | .538 |
| Origin | .188 | .128 | .189 | .320 | .326 | .227 | .355 | .534 |

**Table 3.** ATiSE results on Yago11k and Wiki12k.

| Method | Yago11k | | | | Wiki12k | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| Claude | .171 | .113 | .170 | .290 | .283 | .181 | .318 | .487 |
| Bard | **.176** | **.116** | **.180** | **.299** | **.290** | **.183** | **.331** | **.497** |
| ERNIE | .166 | .106 | .171 | .287 | .275 | .179 | .309 | .461 |
| GPT-3.5 | .170 | .107 | .172 | .304 | .274 | .174 | .310 | .473 |
| Original | .168 | .109 | .169 | .289 | .280 | .175 | .318 | .481 |

### 5.2 Prediction Tasks

This task employed the CyGNet model and RE-NET model.Tables 4 and 5 respectively describe the results of CyGNet and RE-NET on the Yago11k and Wiki12k datasets. In these experiments, Claude, Bard, ERNIE, and GPT-3.5 represent the names of the datasets enhanced by the corresponding large language models, while "Origin" denotes the original dataset.

Overall, the dataset enhanced by the large language models did not perform well on the prediction models CyGNet and RE-NET.Especially in the case of the CyGNet model, its performance showed a noticeable decrease.However, the dataset enhanced by Claude showed a relatively promising performance on the RE-NET model.

Temporal knowledge graph reasoning tasks and prediction tasks have different requirements and characteristics. The distribution of temporal data at different timestamps has a significant impact on prediction tasks. This may be a contributing factor to the suboptimal performance of the data enhanced through data augmentation on prediction tasks.

**Table 4.** CyGNet results on Yago11k and Wiki12k.

| Method | Yago11k | | | | Wiki12k | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| Claude | .627 | .588 | .651 | **.693** | .449 | .404 | .476 | **.530** |
| Bard | .626 | .587 | .650 | .689 | .448 | .404 | .474 | .529 |
| ERNIE | .627 | .587 | .651 | .692 | .448 | .403 | .475 | .529 |
| GPT-3.5 | .626 | .586 | .648 | .692 | .450 | .406 | .477 | .534 |
| Origin | **.634** | **.598** | **.656** | .690 | **.453** | **.411** | **.479** | **.530** |

**Table 5.** RE-NET results on Yago11k and Wiki12k.

| Method | Yago11k | | | | Wiki12k | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| Claude | **.649** | **.632** | **.653** | **.681** | **.522** | **.511** | **.524** | **.540** |
| Bard | .646 | .629 | .651 | .676 | .520 | **.511** | .523 | .538 |
| ERNIE | .647 | .630 | .652 | .676 | .520 | .510 | .521 | .539 |
| GPT-3.5 | .644 | .627 | .647 | .670 | .521 | **.511** | .523 | **.540** |
| Origin | .647 | .631 | .649 | .676 | .520 | .510 | .522 | .538 |

## 6 Conclusions

This paper aims to explore the feasibility and effectiveness of large language models as external knowledge bases for temporal knowledge graph reasoning tasks. By integrating the collaborative mode of large language models and temporal knowledge graphs, we aim to validate the specific impact of this combination on the performance

of temporal knowledge graph reasoning. Through experimental validation, we observed that this collaborative mode had a positive impact on temporal knowledge graph completion tasks. However, its influence on prediction tasks was not significant.

## Acknowledgements

This work was partially supported by the Guangdong Natural Science Foundation (No. 2018A030313777).

## References

1. Leblay, J., Chekol, M. W.: Deriving validity time in knowledge graph. In: Companion Proceedings of The Web Conference 2018, pp. 1771–1776 (2018).
2. Wang, Q., Mao, Z., Wang, B., et al.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering, 29(12), 2724–2743 (2017).
3. Bordes, A., Usunier, N., Garcia-Duran, A., et al.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26 (2013).
4. Yang, B., Yih, W., He, X., et al.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014).
5. Schlichtkrull, M., Kipf, T. N., Bloem, P., et al.: Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, pp. 593-607. Springer International Publishing (2018).
6. Jiang, T., Liu, T., Ge, T., et al.: Towards time-aware knowledge graph completion. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1715-1724 (2016).
7. Bordes, A., Usunier, N., Garcia-Duran, A., et al.: Translating embeddings for modeling multi-relational data. Advances in Neural Information Processing Systems, 26 (2013).
8. Dasgupta, S. S., Ray, S. N., Talukdar, P.: Hyte: Hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp. 2001-2011 (2018).
9. Wang, Z., Zhang, J., Feng, J., et al.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1 (2014).
10. Trivedi, R., Dai, H., Wang, Y., et al.: Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: International Conference on Machine Learning, pp. 3462-3471. PMLR (2017).
11. Ma, Y., Tresp, V., Daxberger, E. A.: Embedding models for episodic knowledge graphs. Journal of Web Semantics, 59, 100490 (2019).
12. García-Durán, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. arXiv preprint arXiv:1809.03202 (2018).
13. Yang, B., Yih, W., He, X., et al.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014).
14. Seo, Y., Defferrard, M., Vandergheynst, P., et al.: Structured sequence modeling with graph convolutional recurrent networks. In: Neural Information Processing: 25th

International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25, pp. 362-373. Springer International Publishing (2018).
15. Trivedi, R., Farajtabar, M., Biswal, P., et al.: Dyrep: Learning representations over dynamic graphs. In: International Conference on Learning Representations (2019).
16. Jin, W., Jiang, H., Qu, M., Chen, T., Zhang, C., Szekely, P., and Ren, X.: Recurrent Event Network: Global Structure Inference over Temporal Knowledge Graph. arXiv preprint arxiv:1904.05530v3 (2019).
17. Schlichtkrull, M., Kipf, T. N., Bloem, P., et al.: Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, pp. 593-607. Springer International Publishing (2018).
18. Bang, Y., Cahyawijaya, S., Lee, N., et al.: A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv preprint arXiv:2302.04023 (2023).
19. Liang, P., Bommasani, R., Lee, T., et al.: Holistic evaluation of language models. arXiv preprint arXiv:2211.09110 (2022).
20. Lee, N., An, N. M., Thorne, J.: Can Large Language Models Infer and Disagree Like Humans? arXiv preprint arXiv:2305.13788 (2023).
21. Frieder, S., Pinchetti, L., Griffiths, R. R., et al.: Mathematical capabilities of chatgpt. Advances in Neural Information Processing Systems, 36 (2024).
22. Pu, D. and Demberg, V.: ChatGPT vs Human-authored Text: Insights into Controllable Text Summarization and Sentence Style Transfer. arXiv:2306.07799 (2023).
23. Gekhman, Z., Herzig, J., Aharoni, R., et al.: Trueteacher: Learning factual consistency evaluation with large language models. arXiv preprint arXiv:2305.11171 (2023).
24. Lai, V. D., Ngo, N. T., Veyseh, A. P. B., et al.: ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. arXiv preprint arXiv:2304.05613 (2023).
25. He, B., Zhou, D., Xiao, J., et al.: Integrating graph contextualized knowledge into pre-trained language models. arXiv preprint arXiv:1912.00147 (2019).
26. Su, Y., Han, X., Zhang, Z., et al.: Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. AI Open, 2, 127-134 (2021).
27. Yu, D., Zhu, C., Yang, Y., et al.: Jaket: Joint pre-training of knowledge graph and language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 10, pp. 11630-11638 (2022).
28. Lin, B. Y., Chen, X., Chen, J., et al.: Kagnet: Knowledge-aware graph networks for commonsense reasoning. arXiv preprint arXiv:1909.02151 (2019).
29. Feng, Y., Chen, X., Lin, B. Y., et al.: Scalable multi-hop relational reasoning for knowledge-aware question answering. arXiv preprint arXiv:2005.00646 (2020).
30. Yasunaga, M., Ren, H., Bosselut, A., et al.: QA-GNN: Reasoning with language models and knowledge graphs for question answering. arXiv preprint arXiv:2104.06378 (2021).
31. Sun, Y., Shi, Q., Qi, L., et al.: JointLK: Joint reasoning with language models and knowledge graphs for commonsense question answering. arXiv preprint arXiv:2112.02732 (2021).
32. Zhang, X., Bosselut, A., Yasunaga, M., et al.: Greaselm: Graph reasoning enhanced language models for question answering. arXiv preprint arXiv:2201.08860 (2022).
33. Vrandečić, D.: Wikidata: A new platform for collaborative data collection. In: Proceedings of the 21st International Conference on World Wide Web, pp. 1063-1064 (2012).
34. Hoffart, J., Berberich, K., Weikum, G.: A spatially and temporally enhanced knowledge base from Wikipedia: YAGO2. Artificial Intelligence (2013).

35. Xu, C., Nayyeri, M., Alkhoury, F., et al.: Temporal knowledge graph embedding model based on additive time series decomposition. arXiv preprint arXiv:1911.07893 (2019).
36. Xu, C., Chen, Y. Y., Nayyeri, M., et al.: Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2569-2578 (2021).
37. Zhu, C., Chen, M., Fan, C., et al.: Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4732-4740 (2021).